

**LAPORAN TUGAS UAS
PEMODELAN & SIMULASI IF-41**

**PENYEBARAN PENYAKIT/VIRUS DENGAN
RANDOM WALK**



Oleh:

DAFFA ULAYYA SUHENDRA - 1301184328

RIZAL MUHAMMAD FAUZAN - 1301180150

**IF 41-GAB 01
PROGRAM STUDI INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
BANDUNG**

2021

DAFTAR ISI

Deskripsi Permasalahan	2
Pemodelan Matematika	2
ALGORITMA	2
Algoritma Program	3
Hasil Simulasi	4
Kesimpulan	5

1. DESKRIPSI PERMASALAHAN

Buatlah simulasi penyebaran penyakit/virus dengan menggunakan Random Walk

4 Arah dengan menggunakan variabel-variabel berikut:

- Jumlah individu: 200
- Rasio individu terinfeksi: 5%
- Probabilitas individu bergerak: 80%
- Waktu pemulihan: 10 hari
- Ukuran ruang simulasi: 20 x 20 unit

1. Challenge 1

- a. Lakukan simulasi dan buatlah plot jumlah individu yang terinfeksi tiap harinya.
- b. Tentukan waktu pemulihan yang diperlukan oleh komunitas tersebut

2. Challenge 2

Buatlah video dari simulasi tersebut dan berikan pembeda (misal: warna) untuk individu yang sehat dan individu yang terinfeksi

2. PEMODELAN MATEMATIKA

Model yang digunakan adalah Random Walk yang menggambarkan jalur yang terdiri dari sukseki langkah acak berturut-turut dalam suatu ruang matematis.

3. ALGORITMA

Dalam simulasi ini, kami menggunakan bahasa pemrograman yang digunakan adalah python

3.1 Algoritma Program

```
1 import numpy as np
2 from matplotlib import animation
3 import matplotlib.pyplot as plt
4
5 #inisialisasiukuranruangsimulasi
6 xmin = 0
7 xmax = 20
8 ymax = 20
9 ymin = 0
10 xrange = xmax - xmin
11 yrange = ymax - ymin
12
13 #inisialisasi variabel
14 iterasi = 120 #asumsi iterasi untuk melakukan percobaan
15 sum_individu = 200 #total jumlah individu
16 ratio_infection = 0.05 #rasio individu yang terinfeksi
17 time_recov = 10 #waktu pemulihan
18 probs_move = 0.80 #probabilitas individu bergerak
19
20 #inisialisasi posisi people
21 posisi_x = np.zeros((sum_individu, iterasi))
22 posisi_y = np.zeros((sum_individu, iterasi))
23
24 #status
25 belum_infected = -1
26 kebal = 0
27
28 #infected people
29 inf_people = np.zeros((sum_individu, iterasi))
30 infcount = np.zeros(iterasi)
31 hari = np.arange(1, (iterasi-1), 1)
32 count_day = np.zeros(iterasi)
33
34 #fungsi untuk animasi gerakan
35 def animate(i):
36     plt.set_data(count_day[:i+1], infcount[:i+1])
37     for i in range(sum_individu):
38         create('p'+str(i+1)), set_data(posisi_x[i][i], posisi_y[i][i])
39         if (inf_people[i][i] > 0):
40             create('p'+str(i+1)), set_color('red') #memberi warna red kepada yang individu terinfeksi
41         elif (inf_people[i][i] < 0):
42             create('p'+str(i+1)), set_color('blue') #memberi warna blue kepada yang belum terinfeksi
43         else:
44             create('p'+str(i+1)), set_color('blue') #memberi warna blue lagi untuk individu yang sudah pulih
45     return ((create('p'+str(k+1)) for k in range(sum_individu)), (p12))
46
47
48 #fungsi untuk melakukan update status dan bisa menambahkan waktu recovery sehingga kita bisa mengidentifikasi nantinya berapa banyak jumlah pasien
49 def update_status(inf_people, sum_individu, time_recov, j, posisi_x, posisi_y):
50     for a in range(sum_individu):
51         if inf_people[a][j] == 0:
52             for b in range(sum_individu):
53                 if (posisi_x[a][j] == posisi_x[b][j] and posisi_y[a][j] == posisi_y[b][j]):
54                     if (inf_people[b][j] == 1):
55                         inf_people[a][j] = inf_people[b][j] + time_recov
56
57 #program utama
58 for j in range(0, iterasi):
59     for i in range(0, sum_individu):
60         if (j == 0): #test posisi mula individu
61             posisi_x[i][0] = np.random.randint(low=xmin, high=xmax)
62             posisi_y[i][0] = np.random.randint(low=ymin, high=ymax)
63             ran_infected = np.random.rand()
64             if (ratio_infection > ran_infected): #memberikan time_recov kepada orang yang terkena infected di awal
65                 inf_people[i][0] = time_recov
66             else:
67                 inf_people[i][0] = belum_infected
68
69         if (inf_people[i][j-1] kebal): #tahap recovery menuju kebal
70             inf_people[i][j] = inf_people[i][j-1] - 1
71         if (inf_people[i][j-1] < 0): #belum infected
72             inf_people[i][j] = belum_infected
73         if (inf_people[i][j-1] kebal): #mengingat kasus ketika individu masih terinfeksi
74             infcount[j] = infcount[j] + 1
75         ran_move = np.random.rand()
76         if (probs_move > ran_move): #randommove sesuai dengan nilai probabilitas
77             rand = np.random.rand()
78             if rand < 0.25:
79                 posisi_x[i][j] = posisi_x[i][j-1] + 1
80                 posisi_y[i][j] = posisi_y[i][j-1]
81             elif rand < 0.50:
82                 posisi_x[i][j] = posisi_x[i][j-1]
83                 posisi_y[i][j] = posisi_y[i][j-1] + 1
84             elif rand < 0.75:
85                 posisi_x[i][j] = posisi_x[i][j-1] - 1
86                 posisi_y[i][j] = posisi_y[i][j-1]
87             else:
88                 posisi_x[i][j] = posisi_x[i][j-1]
89                 posisi_y[i][j] = posisi_y[i][j-1] + 1
90         else: #jika tidak bergerak akan tetap dengan posisi sebelumnya
91             posisi_x[i][j] = posisi_x[i][j-1]
92             posisi_y[i][j] = posisi_y[i][j-1]
93
94
95 #bereski posisi dengan PK
96 if posisi_x[i][j] > xmax:
97     posisi_x[i][j] = posisi_x[i][j] - xrange
98 elif posisi_x[i][j] < 0:
99     posisi_x[i][j] = posisi_x[i][j] + xrange
100 elif posisi_y[i][j] > ymax:
101     posisi_y[i][j] = posisi_y[i][j] - yrange
102 elif posisi_y[i][j] < 0:
103     posisi_y[i][j] = posisi_y[i][j] + yrange
104
105 update_status(inf_people, sum_individu, time_recov, j, posisi_x, posisi_y)
106 sum_terinfeksi = infcount[j]
107 totalday = j
108 count_day[i] = j
109 print('Hari ke', j, sum_terinfeksi, 'orang')
110 if (sum_terinfeksi == 0 and j > 0): #iterasi akan berhenti ketika jumlah terinfeksi sudah menemukan angka 0
111     break
112
113 print('Total Hari Kasus', totalday)
114
115 #create figure animation
116 fig = plt.figure(num=0, figsize=(15, 12))
117 create = dict()
118 ax01 = plt.subplot2grid((2, 2), (0, 0))
119 plt.xlabel('x')
120 plt.ylabel('y')
121 ax02 = plt.subplot2grid((2, 2), (0, 1))
122 plt.xlabel('days')
123 plt.ylabel('Number of infected people')
124
125 p12, = ax02.plot(count_day, infcount, '-b')
126
127 for i in range(sum_individu):
128     create['p'+str(i+1)], = ax01.plot(posisi_x[i][0], posisi_y[i][0], 'o', c='b')
129     if (inf_people[i][0] < 0):
130         create['p'+str(i+1)], = ax01.plot(posisi_x[i][0], posisi_y[i][0], 'o', c='r')
131     elif (inf_people[i][0] < 0):
132         create['p'+str(i+1)], = ax01.plot(posisi_x[i][0], posisi_y[i][0], 'o', c='b')
133
134 anim = animation.FuncAnimation(fig, animate, frames=400, interval=150, blit=False, repeat=True)
135 plt.show()
```

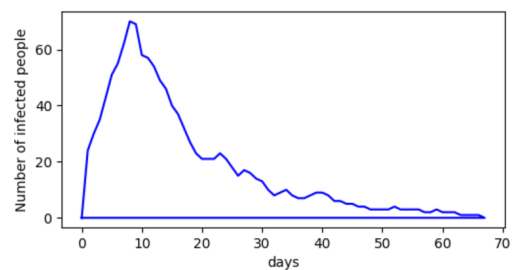
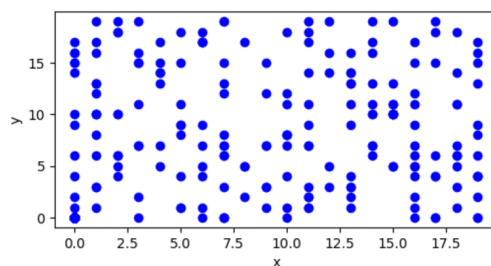
3.2 Hasil Simulasi

Berikut hasil simulasi dari jumlah individu yang terinfeksi tiap harinya, dan waktu pemulihan yang diperlukan oleh komunitas tersebut.

```
D:\Muliha 2021\WosilTubes\py CodeProgram.py
Hari ke 0 0.0 orang
Hari ke 1 24.0 orang
Hari ke 2 30.0 orang
Hari ke 3 35.0 orang
Hari ke 4 43.0 orang
Hari ke 5 51.0 orang
Hari ke 6 55.0 orang
Hari ke 7 62.0 orang
Hari ke 8 70.0 orang
Hari ke 9 69.0 orang
Hari ke 10 55.0 orang
Hari ke 11 57.0 orang
Hari ke 12 54.0 orang
Hari ke 13 49.0 orang
Hari ke 14 46.0 orang
Hari ke 15 40.0 orang
Hari ke 16 37.0 orang
Hari ke 17 32.0 orang
Hari ke 18 27.0 orang
Hari ke 19 23.0 orang
Hari ke 20 21.0 orang
Hari ke 21 21.0 orang
Hari ke 22 21.0 orang
Hari ke 23 23.0 orang
Hari ke 24 21.0 orang
Hari ke 25 18.0 orang
Hari ke 26 15.0 orang
Hari ke 27 17.0 orang
Hari ke 28 16.0 orang
Hari ke 29 14.0 orang
Hari ke 30 13.0 orang
Hari ke 31 10.0 orang
Hari ke 32 8.0 orang
Hari ke 33 9.0 orang
Hari ke 34 10.0 orang
Hari ke 35 8.0 orang
```

```
Hari ke 35 8.0 orang
Hari ke 36 7.0 orang
Hari ke 37 7.0 orang
Hari ke 38 8.0 orang
Hari ke 39 9.0 orang
Hari ke 40 9.0 orang
Hari ke 41 8.0 orang
Hari ke 42 6.0 orang
Hari ke 43 6.0 orang
Hari ke 44 5.0 orang
Hari ke 45 5.0 orang
Hari ke 46 4.0 orang
Hari ke 47 4.0 orang
Hari ke 48 3.0 orang
Hari ke 49 3.0 orang
Hari ke 50 3.0 orang
Hari ke 51 3.0 orang
Hari ke 52 4.0 orang
Hari ke 53 3.0 orang
Hari ke 54 3.0 orang
Hari ke 55 3.0 orang
Hari ke 56 3.0 orang
Hari ke 57 2.0 orang
Hari ke 58 2.0 orang
Hari ke 59 3.0 orang
Hari ke 60 2.0 orang
Hari ke 61 2.0 orang
Hari ke 62 2.0 orang
Hari ke 63 1.0 orang
Hari ke 64 1.0 orang
Hari ke 65 1.0 orang
Hari ke 66 1.0 orang
Hari ke 67 0.0 orang
Total Hari Kasus 67
```

Contoh grafik dari hasil simulasi



4. KESIMPULAN

Proses penyebaran suatu penyakit/virus dengan iterasi 120 ini menyebar lebih cepat di awal munculnya penyakit/virus itu, tetapi semakin hari penyebarannya menurun karena orang yang sudah terkena penyakit/virus sudah melalui proses pemulihan.